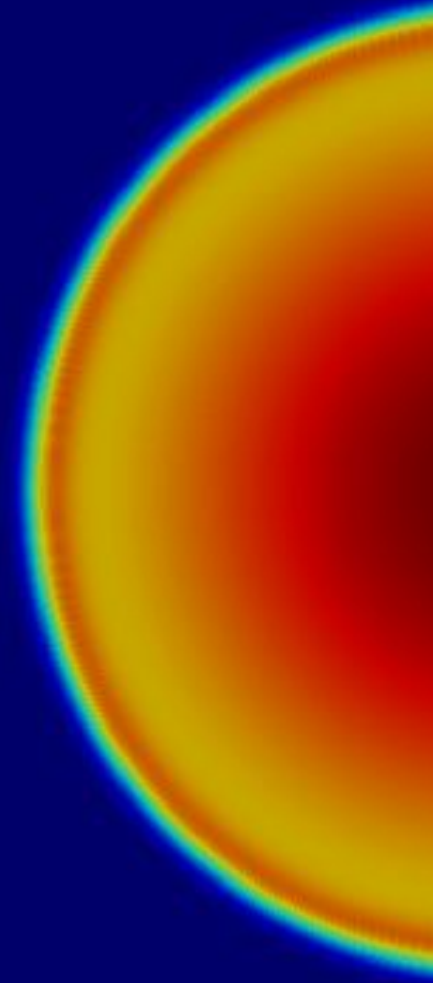


Summer of RNTuple

GSoC with the ROOT IO team

Max Orok

Mentored by Jakob Blomer & Philippe Canal



Project

“Fast Merging of RNTuple Data Sets” ([link](#))

Next generation file format for physics data

Project objective: develop a fast merge algorithm to combine files

Results

1. Low-level merge functions
2. Asynchronous IO using `io_uring`
3. RNTuple library ergonomics

Merging (1)

Reading and writing an RNTuple file:

1. Find the byte range corresponding to your data
2. Uncompress the bytes
3. Unpack the bytes to the C++ representation
4. ...
5. Repack the bytes
6. Recompress
7. Write the byte range to storage

Merging (2)

Fast merging:

1. Find the byte range corresponding to your data
2. Write the byte range to storage

Pull requests:

- #6105 – low level reading and writing
- #6101 – communicate with the hadd tool

Async IO

Investigated the new **io-uring** asynchronous IO kernel interface

Preliminary results show a 4-5x speedup on SSD read throughput (no changes to end user code!)

Pull requests:

- [#5919](#) – add io-uring build option
- [#6162](#) – implement asynchronous reads

Thank you

Thank you to my mentors, Jakob Blomer and Philippe Canal. Thank you also to the GSoC team at CERN.

Ergonomics

Making the library easier to use

Example: adding fields to an RNTuple schema

Pull requests:

- [#5934](#) – add field builder interface
- [#5848](#) – iterate over field metadata

Ergonomics

Before:

```
descBuilder.AddField(  
    1,  
    RNTupleVersion(),  
    RNTupleVersion(),  
    "someField",  
    "",  
    0,  
    ENTupleStructure :: kRecord  
);
```

Ergonomics

After:

```
descBuilder.AddField(RDanglingFieldDescriptor()  
    .FieldId(1)  
    .FieldName("someField")  
    .Structure(ENTupleStructure::kRecord)  
    .MakeDescriptor()  
    .Unwrap());
```